EXPLORATIVE ANALYSIS OF GRAPH PYRAMIDS USING INTERACTIVE VISUALIZATION TECHNIQUES

Andreas Kerren
Computer Science Department
University of Kaiserslautern
P.O.Box 3049
D-67653 Kaiserslautern, Germany
email: kerren@acm.org

ABSTRACT
Hierarchies of plane graphs, called graph pyramids, can be used for collecting, storing, and analyzing geographical information based on images or other input data. The visualization of graph pyramids facilitates studies about their structure, such as their vertex distribution or height in relation of a specific input image. Thus, a researcher can debug contraction processes and ask for statistical information. Furthermore, it improves the better understanding of geographical data, like landscape properties or thematical maps. In this paper, we present an interactive visualization tool that supports several coordinated views on graph pyramids, subpyramids, level graphs, thematical maps, etc. We discuss a brief case study on the basis of a small sample image and conclude the paper with future work.

KEY WORDS
Visualization Software, Graph Pyramids, HCI, Environmental Sciences

1 Introduction

Graph pyramids [1] (synonymous with *image pyramids*) store hierarchies of graphs and the linkage between consecutive graph levels. They allow multiple scale and abstraction of the description of images. In general, they can be built from images or pre-segmented graphs in GML format [2] which are extended with any additional information. Each node of the base level represents a pixel with a specific color value. This is the node's attribute. The edges form a regular grid. In case of pre-segmented input graphs, the base level corresponds to the elements of the input graph. Additionally, nodes and edges can be annotated with user-defined attributes, e.g., with colors or properties such as the degree of border precision between segmented image areas. Figure 1(a) shows a small example of a graph pyramid. In *regular* image pyramids, the proportion of the number of nodes at two levels l and $l+1$ is given by a fixed constant reduction factor $r > 1$. This strict property is softened by *irregular* pyramids where the hierarchical structure is not known from the first.

By contracting edges within one level of the pyramid an algorithm, called *Dual Graph Contraction* (DGC), computes the graph of the next higher level. Edge contraction is a process of collapsing the edge $e = (u, v)$ and melting the nodes u and v, i.e. one node (the surviver) remains and the other node is removed. The surviver adopts the edges that were incident to the removed node and its attributes are a combination of those from u, v, and e. DGC is specified by disjoint *contraction kernels* (see Figure 1(b)), each defining a possibly empty set of non-surviving nodes and the surviver. The selection of a contraction kernel depends on edge and node attributes. They are described by so-called *contraction rules* which are highly dependent on the application area. The visualization of these contraction processes is an important issue because it illustrates the core of this kind of *meta-segmentation*. See the article of Kropatsch [1] for further details on DGC algorithms and image pyramids.

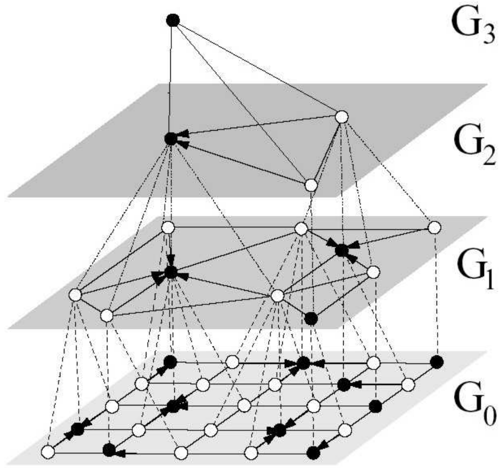
1.1 Applications of Graph Pyramids

There are several application areas of (irregular) graph pyramids. Image segmentation or feature detection are well-known examples [5, 6]. Furthermore, Haxhimusa and Kropatsch [7] discuss the hierarchical partitioning of images using pairwise similarity functions on graph-based representations of images. Here, each level represents a pixel partitioning into cells.

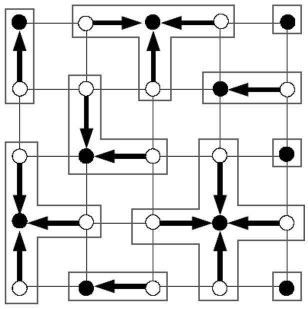
Another application is the automatical determination of landscapes and their properties in Landscape Ecology. Such properties can be very complex and interdependent, for example warmth or humidity in a special area. A landscape itself can consist of several objects, such as lakes or forests. To determine the type of a landscape, we also need information about the size, the shape, as well as some context data of these landscape objects. Thus, a large forest without any streets in alpine scenery could be an interesting wildlife habitat. In the project GEOGRAPH[1], the automatical determination of primitive landscape elements and types is done by segmentation on the basis of images (e.g., satellite images or aerial photography). Graph-based segmentation techniques use node and edge informa-

[1]This research has been supported by the Austrian Science Fund (FWF) under grant number P14662-N04.

Proceedings of the 5th IASTED International Conference on Visualization, Imaging, and Image Processing, VIIP 2005, pages 685-690, Benidorm, Spain, 2005.



(a) Small (irregular) graph pyramid



(b) Contraction kernels

Figure 1. Graph pyramid and contraction kernels (taken from [3] and [4]) are based on a 5×5 pixel image. Directed edges represent the contraction processes, white nodes are collapsed in the next upper level, and dotted edges between the levels symbolize together with the connected nodes all contraction kernels.

tion to perform the meta-segmentation: Each object (landscape area) in a level can be regarded as node in a graph. Edges between nodes indicate the neighborhood relations within that level. The identification of meaningful contraction rules is also part of the project and ongoing research. The result is a graph pyramid that we can use as data structure for retrieving geographical information. Each pyramid level represents a kind of abstract map (land cover maps) of the input image: Lower levels describe more concrete maps and higher levels represent a higher level of abstraction. In the following, we call these maps *thematical maps* due to the possibility to generate them using additional information.

The remainder of this paper is organized as follows: Next, our visualization approach is presented including a description of supported graphical views and navigation techniques. A case study of our tool is shown in Section 3. We conclude with Section 4 and give an outline of future work.

2 Visualization of Graph Pyramids

An appropriate visualization of graph pyramids and related information are important aspects to learn more about the pyramid structure, relationships between different levels, contraction processes, etc. In this context, the visualization of the whole graph pyramid is well suited for navigation purposes. All needed information is stored in this data structure and there is a clear and logical connection between their components. For example, the user can select a special pyramid level and the visualization system opens a view with the represented thematical map computed by a down projection to the base level, see Section 2.1.4. So, the user can easily develop new contraction rules and verify them with the help of the visualization. He/she can analyze the use and computation of contraction kernels level by level as well as the resulting thematical maps. As far as we know, our visualization tool is the first one in the area of interactive visualization of graph pyramids. A good visualization tool should facilitate the understanding of the general structure of the pyramid itself, the contraction rule's effects on the pyramid structure, the verification of the contraction rule set and definitions, the correlation between pyramid levels and thematical maps, the role of node and edge attributes, related statistical information (e.g. its node distribution or reduction factor), and pyramid comparisons.

2.1 Graphical Views

Our visualization tool DGCVis supports currently four different graphical views (2D and 3D) on the pyramid structure as well as related data. Each individual view was designed to fulfill different visualization needs. Although we have to visualize graph structures this is not a traditional graph drawing problem (cp. [8]) because all node positions result from the segmentation and contraction processes. Note that the screenshot examples only show visualizations based on very small input data.

2.1.1 Graph Pyramid View (GPV)

This central 3D view presents three different subviews closely connected with each other. It can be used for common navigation purposes since graph pyramids are the central structures that store all information. The GPV represents a so-called Focus&Context technique (see [9, 10] for example): Users can focus on a specific level or subtree whereas the context of this focus is displayed in the Pyramid Subview:

- *Pyramid Subview:* The Pyramid Subview gives an overview of the data structure, i.e., each plane graph is drawn level by level. The graph nodes are visualized by colored spheres while the edges are thin cylinders, each connecting two spheres. Light green lines which connect two graphs of different hierarchies represent

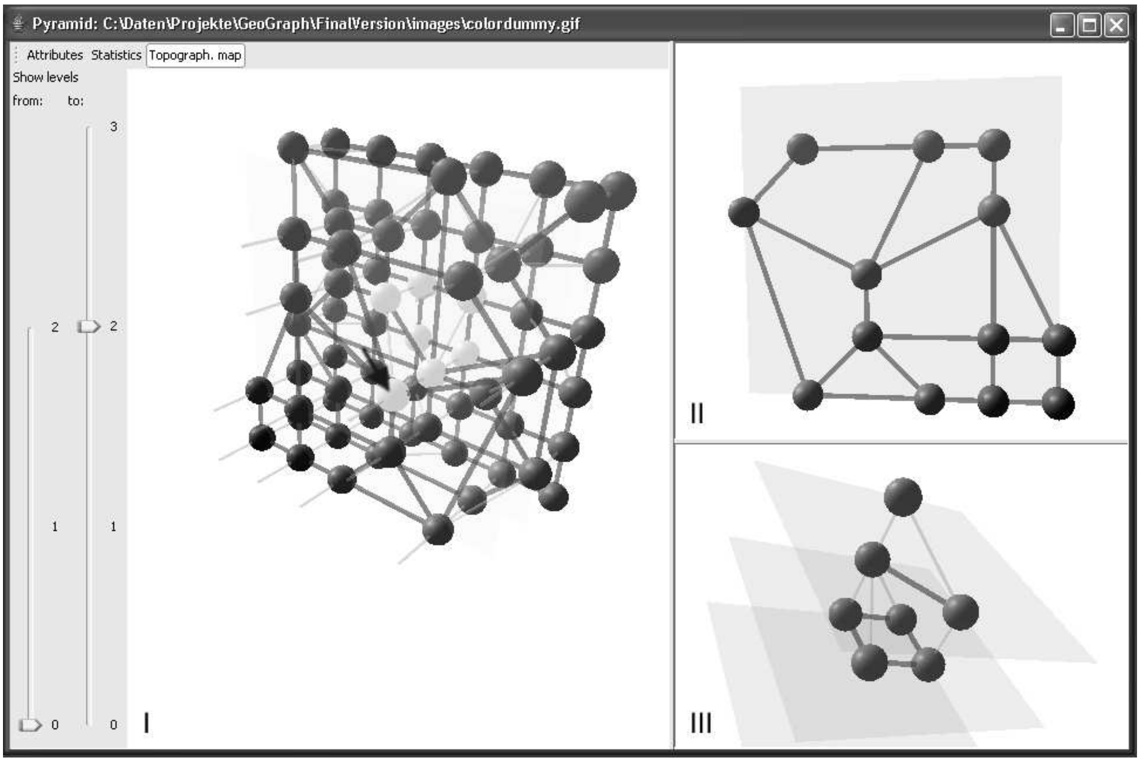


Figure 2. Graph Pyramid View composed of Pyramid (I), Level (II), and Subtree (III) Subview.

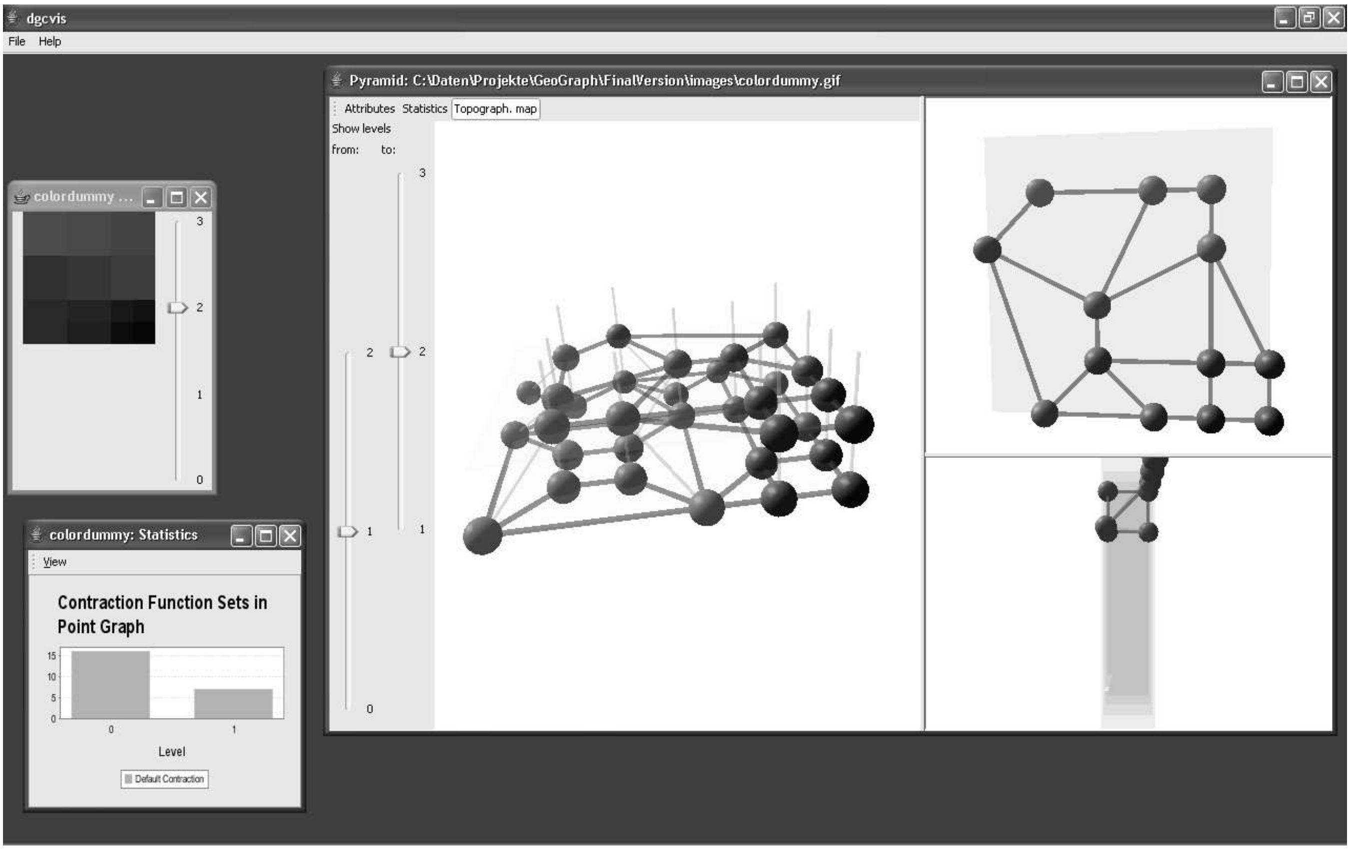


Figure 3. DGCVis opened with three different views and based on a 6×6 pixel color sample image.

the contraction kernels, see Figure 2 (Part I). This figure only shows a small example pyramid, bigger images produce bigger graph pyramids with much more visualization objects. In practice, this visualization of a complete graph pyramid could be too complex for users to get deeper information. For this reason, our prototype implementation supports a visualization of separate pyramid levels which are selected by choosing a level interval with the help of two sliders.

- *Level Subview:* This part of the GPV shows the actual selected level, see Figure 2 (Part II). There are the same navigation facilities as in the Pyramid Subview. From the user's point of view, the visual comparison between the level graph and the thematical map visualization (see Subsection 2.1.4) is very important. Thus, he/she can see the mapping of the graph nodes and the appropriate image region of the computed thematical map.

- *Subtree Subview:* The black arrow in the Pyramid Subview of Figure 2 (Part I) symbolizes a specific node selection at the third pyramid level. Then, the connected subtree is computed and placed in this third view. The subtree computation starts with the selected node and is linked via contraction kernels to the lower levels. These connections are traced and the resulting tree is copied into a new pyramid. In this way, the presentation of the whole data structure is reduced to the level, subtree, and node of interest as the reader can see in Figure 2 (Part III).

All elements of the GPV can be grasped and manipulated. Users can look at the pyramid from every possible viewpoint. They can move (pan) the pyramid or level graph, rotate it and zoom in and out of it using the mouse. In general, DGCVis supports two different node selection types: attributes selection and subtree selection. Both selection types exist side by side. Selected nodes are drawn with different colors (light green, yellow or a mixture of both if a node is both subtree and attributes selected) depending on their selection types.

2.1.2 Attribute View (AV)

In the Attribute View, users can hierarchically navigate through a so-called *TreeView* with all the attribute list types existing in the graph pyramid. Those attribute list types are grouped by level and then in the next hierarchy level by node or edge. Right now, we primarily work only with the color attribute because there are not any appropriate input examples yet. However, the system can work with an unlimited number of different attributes, such as humidity, population density, vegetation, radiation of any kind, pH-value, occurrence of special animals, etc.

2.1.3 Statistics View (SV)

Global statistics are shown as chart diagrams within a separate window. At present, the statistical information a user can obtain are distributions of point nodes, face nodes, and quadedges, attribute list types of point nodes, face nodes, and quadedges, as well as contraction function sets in point and face graphs, each over levels. Figure 3, bottom left-hand corner, shows a bar chart of contraction function sets in the point graph.

2.1.4 Thematical Map View (TMV)

This 2D view shows the selected level's projection to the input image (down projection) on which the graph pyramid is based. Among other things, attributes of each node at the level (for example the RGB value) are forwarded to nodes of the base level along the tree paths built from the contraction kernels. In case of GML graphs as input, the down projection is performed with the help of a path per node for the specification of the area shape that is represented by the node. This information is part of the used GML specification. TMV contains a slider to change the selected level, see the screenshot in Figure 3, top left-hand corner. If there are selected nodes in the Pyramid View, then the TMV displays the down projection only on the basis of the selected nodes at the level selection.

3 Case Study

The Pyramid Subview in Figure 4 shows a graph pyramid of height 11 with a medium reduction factor. Several nodes are present at all pyramid levels, i.e., they have survived all contraction processes. The Subtree Subview of this screenshot displays the selected subtree (see black arrow) which has several contractions within the first four levels. On the left, the TMV shows the down projection of the selected node. Here, the user can study the results of the contraction processes very well. He/she can see which parts of the input image collapse to a specific node in the pyramid visualization. If this is not correct then it is possible to identify the faulty contraction rule. After fixing the problem, the user could compute and visualize a modified graph pyramid and check the contraction rule set again.

The graph of the base level in Figure 5(a) (same input as in Figure 4) forms a regular grid as described in Section 1. Then, nodes and edges are reduced by the DGC algorithm level by level. The node distribution in Level 3 (part (c)) shows a center node and four node accumulations at each corner. Thus, the used contraction rule set seems to be not bad for this kind of sample images because the accumulations represent the four numbers and the center node represents the biggest part of the white background of the input image. However, an ideal solution would be a graph with five nodes: one for the background and four for the numbers. This quality is only achieved for the numbers '3'

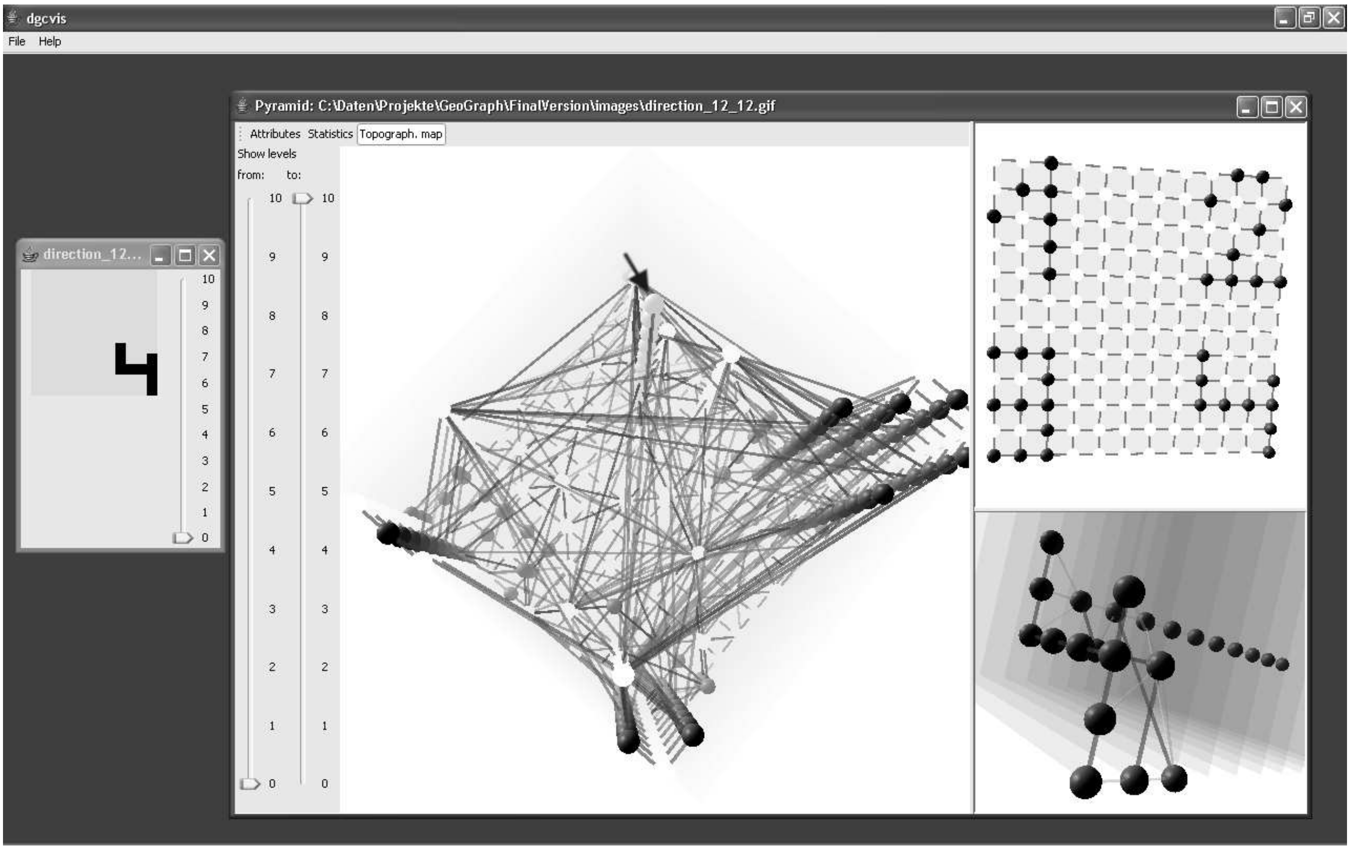
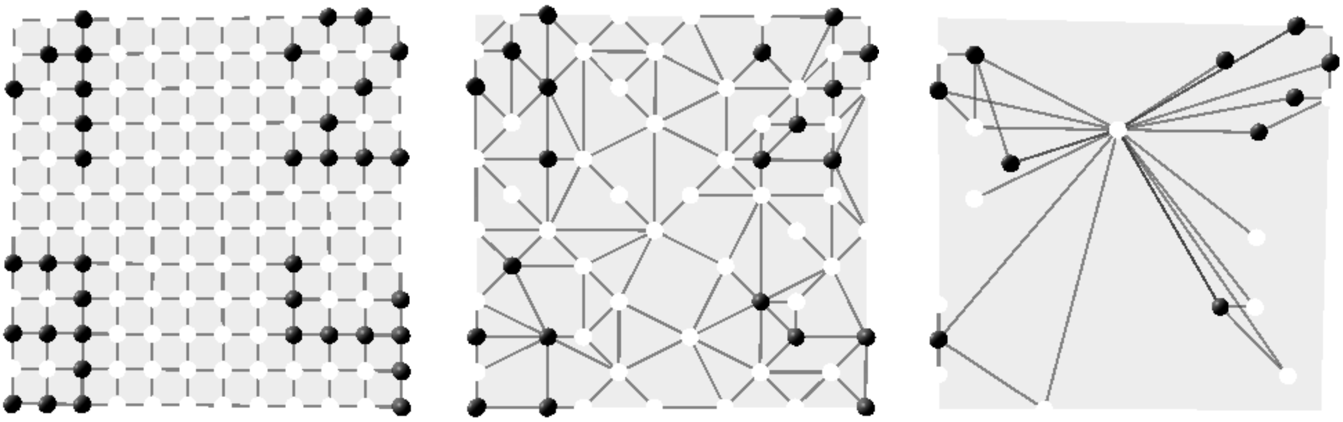


Figure 4. Input of this DGCVis session is a small B/W image of size 12×12 pixels:

1	2
3	4



(a) Level 0 (b) Level 1 (c) Level 3

Figure 5. Level Subviews of three lower pyramid levels.

and '4' in higher levels of our example. More case studies based on an older version of DGCVis and details about implementation and performance aspects can be found in paper [11].

4 Conclusions and Future Work

In this paper, a novel approach for the interactive visualization of graph pyramids was presented. The visualization covers several visualization needs of researchers working on pattern recognition algorithms or on application areas. All implemented views offer many interaction and exploration possibilities to discover new correlations between contraction rules, thematical maps, and the graph pyramid's structure. By using a MVC architecture it is easy to extend DGCVis with additional views.

There are a lot of challenging problems in this area that we plan to solve, e.g., interactive input of contraction rules from the visualization, visualization of graph pyramid changes through movements of objects within an image sequence, or the visualization of pyramid comparisons. In addition, we have to find better navigation solutions and an advanced Focus&Context approach. This will be our main research aim because an extensive study of the existing literature yielded no applicable results for this kind of problems.

Acknowledgements I would like to thank Florian Breier, Philip Kügler, and Michael Schreyer for implementing the DGCVis tool as well as the DGClib interface. Furthermore, I wish to thank the members of the Pattern Recognition and Image Processing Group at the TU Vienna for their useful ideas and support.

References

[1] W.G. Kropatsch, Building Irregular Pyramids by Dual Graph Contraction, *IEE-Proc. Vision, Image and Signal Processing*, 142(6), 1995, 366–374.

[2] M. Himsolt, GML – Graph Modelling Language, `www.uni-passau.de/Graphlet/GML`, 1997.

[3] Y. Haxhimusa, R. Glanz, M. Saib, G. Langs, and W.G. Kropatsch, Logarithmic Tapering Graph Paramid, in L. Van Gool (Ed.), *Proceedings of the 24th DAGM Symposium 2002*, Zürich, Swiss, 2002, 117–124.

[4] W.G. Kropatsch, Y. Haxhimusa, Z. Pizlo, and G. Langs, Vision Pyramids that do not Grow too High, *Pattern Recognition Letters*, 26(3), 2005, 319–337.

[5] L. Guigues, L.M. Herve, and J. P. Cocquerez, The Hierarchy of the Cocoons of a Graph and its Application to Image Segmentation, *Pattern Recognition Letters*, 24(8), 2003, 1059–1066.

[6] P. Meer, D. Mintz, A. Montanvert, and A. Rosenfeld, Consensus Vision, in *Proceedings of the AAAI-90 Workshop on Qualitative Vision*, Boston, MA, USA, 1990, 111–115.

[7] Y. Haxhimusa and W.G. Kropatsch, Hierarchical Image Partitioning with Dual Graph Contraction, in B. Michaelis and G. Krell (Eds.), *Proceedings of the 25th DAGM Symposium 2003*, Magdeburg, Germany, 2003, 338–345.

[8] G. Di Battista, P. Eades, R. Tamassia, and I.G. Tollis, *Graph Drawing – Algorithms for the Visualization of Graphs*, Prentice Hall, 1999.

[9] I. Herman, G. Melançon, and M.S. Marshall, Graph Visualization and Navigation in Information Visualization: A Survey, *IEEE Transactions on Visualization and Computer Graphics*, 6(1), 2000, 24–43.

[10] S.K. Card, J.D. Mackinlay, and B. Shneiderman (Eds.), *Readings in Information Visualization: Using Vision to Think*, Morgan Kaufmann, 1999.

[11] A. Kerren, F. Breier, and P. Kügler, DGCVis: An Exploratory 3D Visualization of Graph Pyramids, in J. C. Roberts (Ed.), *Proceedings of the 2nd International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV '04)*, London, UK, 2004, 73–83.